NASA TECHNICAL MEMORANDUM

NASA TM X-2181

NASA TM X-2181

PREJUD – A COMPUTER CODE FOR THE PRELIMINARY ANALYSIS OF TWO-DIMENSIONAL PULSE HEIGHT ANALYZER DATA

by Thor T. Semler

Lewis Research Center
Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1971

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM X-2181		

4. Title and Subtitle	5. Report Date
PREJUD - A COMPUTER CODE FOR THE PRELIMINARY ANALYSIS OF TWO-DIMENSIONAL PULSE HEIGHT ANALYZER DATA	February 1971
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Thor T. Semler	E-5824
	10. Work Unit No.
	129-02

9. Performing Organization Name and Address	11. Contract or Grant No.
Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer program for the processing of two-dimensional analyzer pulse height data, which provides an automated separation of gamma counts from neutron counts, and produces data in a format suitable for input to a spectrum unfolding code, for example FERDOR, is described. A strategy for the separation of gamma and neutron counts is detailed. Data smoothing techniques and a spline interpolation technique are described in some detail.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Pulse height analysis Computer program Neutron spectra Spline fit Data smoothing	Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	31	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

PREJUD - A COMPUTER CODE FOR THE PRELIMINARY ANALYSIS OF TWO-DIMENSIONAL PULSE HEIGHT ANALYZER DATA

by Thor T. Semler

Lewis Research Center

SUMMARY

A computer program, PREJUD, has been developed for the processing of two-dimensional analyzer, pulse height data. The program provides an automated separation of gamma counts from neutron counts and produces data in a format suitable for input into spectrum unfolding codes, for example, FERDOR.

A strategy for the separation of gamma and neutron counts is detailed. Data smoothing techniques and a spline interpolation technique are described in some detail.

An example of the unfolding of data from a plutonium-beryllium source is shown to illustrate the use of the code.

INTRODUCTION

It has been found experimentally that the overall shapes of the scintillation pulse induced in trans-stilbene and other organic scintillators by gamma-rays and fast neutrons differ (ref. 1). This effect has been used as the basis of a technique (pulse shape discrimination) demonstrated by Brooks (ref. 2) to use organic scintillators to discriminate between gamma-ray and neutron induced scintillations. Details of the circuitry developed for pulse shape discrimination can be found in reference 1.

At the Lewis Research Center a modified Owen pulse shape discrimination circuit has been used (ref. 3). The output of the pulse shape circuit and a linear output from the photomultiplier tube are amplified and recorded in a 4096 channel two-dimensional analyzer. The proton recoil and Compton recoil spectra are measured and the FERDOR code (ref. 4) is used to unfold the proton recoil spectra.

Generally, the preparation of data for the FERDOR code has been done manually (ref. 5). The PREJUD (preliminary adjustment of data) code described herein allows one to prepare suitable input to FERDOR, or other proton recoil spectra unfolding codes,

from the multichannel analyzer data with up to four different gain settings and a minimum of inspection.

PROBLEM ANALYSIS

The data from a single experiment, one gain setting and one counting time, are provided as a two-dimensional array, 64 by 64, of count rate as a function of energy. The results of a well resolved experiment for one gain setting are shown in figure 1. The first phase of the data analysis is the separation of the gamma induced counts from the neutron induced counts to provide a set of neutron counts as a function of energy. The technique used is described in the next section, MINIMUM FINDING STRATEGY. Next the process is repeated for the background, if any, which is then subtracted from the data. If more than one gain setting has been used, the previous steps are repeated to provide several sets of neutron induced counts as a function of energy. Next these sets of data are smoothed to remove some of the experimental scatter. The smoothing techniques are derived in the section SMOOTHING TECHNIQUES. The sets of smoothed data are then normalized to the same gain setting and merged. The merged data are smoothed, and then these results are interpolated and extrapolated at specified energy points so as to provide input to the code FERDOR (ref. 4). The interpolation is described in the section SPLINE INTERPOLATION. The extrapolation techniques (low energy and high energy) are given in the section EXTRAPOLATION OF DATA. The error estimate given these interpolated and extrapolated values is detailed in the ERROR ANALYSIS section. An example of the use of the codes PREJUD and FREDOR is shown in the RESULTS section. The data input for the program is given in appendix A and the program listing is given in appendix B.

MINIMUM FINDING STRATEGY

As indicated in the section on PROBLEM ANALYSIS, the first task of the code is the separation of counts into electron recoil counts caused by gamma rays and proton recoil counts caused by neutrons. This is accomplished by searching along rows of equal energy for the first minimum and summing to the left for electron recoil counts and to the right for proton recoils. Half of the minimum is added to each sum. An idealized situation is shown in figure 2, where the minimum between the Compton recoil spectra and the proton recoil spectra is easily discerned. The technique used to separate the two sets of counts is described in this section and shown in flow chart form in figure 3.

(1) A row of counts A_i ($i = I_g$, 64) corresponding to the same energy is read into the

minimum finding subroutine, where I_g is the number of columns ignored, if any.

(2) If a minimum i_{min} has been found in an earlier row of this data set, start the minimum search five positions to the left of the earlier minimum and then go to step (12).

(3) If the magnitude of any of the first four data cells is greater than 100, go to step (11).

(4) Check for zeros in the first three data cells. If all are zero, go to step (9).

(5) Set i to two larger than the first data cell in this row. Check for two sequentially zero data cells. If they are found, go to step (7).

(6) Increment i counter by one and return to step (5).

(7) Subtract 1 from i and sum from left to i for electron recoil counts and from i to right for proton recoil counts.

(8) Set i_{min} equal to i and return to calling routine.

(9) Sum entire row since there are no electron recoil counts and set equal to proton recoil counts.

(10) Set i_{min} equal to first data cell i and return to calling routine.

(11) Set i equal to one larger than first data cell i.

(12) Check for condition $A_{i-1} > A_i < A_{i+1}$; if this condition is satisfied, go to step (17).

(13) Check for condition $A_{i-2} > A_{i-1} = A_i < A_{i+1}$; if this condition is satisfied, go to step (17).

(14) Check for condition $A_{i-3} > A_{i-2} = A_{i-1} = A_i < A_{i+1}$; if this condition is satisfied, go to step (17).

(15) Check for $A_i = 0$; if it is true, set $i_{min} = i$ and go to step (19).

(16) Increment i by 1 and go to step (12).

(17) If $A_i > (A_{i+1} - 3\sqrt{A_{i+1}})$, go to step (22).

(18) If $A_i > A_{i+2}$, go to step (16).

(19) Set $i_{min} = i$ and perform double summing.

(20) If the sum of the proton recoil counts is less than seven times the counts in $A_{i,min}$ or if the ratio of electron recoil counts over proton recoil counts is greater than 10^4, the sums and i_{min} are set equal to zero.

(21) Return to the calling routine.

(22) If any of the differences $(A_{i+4} - A_i)$, $(A_{i+3} - A_i)$, or $(A_{i+2} - A_i)$ is negative, a local minimum has not been found and the code returns to step (16).

SMOOTHING TECHNIQUES

Once the data have been resolved and stored in bins corresponding to the channel energy, they may be smoothed prior to normalization and spline fitting. Comment cards

are inserted in the code listing (appendix B) indicating how one may either smooth the data more or may eliminate most of the smoothing. A five point least squares fit of the log counts has been chosen to smooth these data. The log counts have been chosen so as not to accentuate large count rates in the smoothing. Since the curvature may well be changing considerably between data points, a five point least squares parabola has been constructed to smooth the equal interval data.

Begin by taking five points centered on (x_0, y_0), that is, (x_{-2}, y_{-2}), (x_{-1}, y_{-1}), (x_0, y_0), (x_1, y_1), and (x_2, y_2), and transform the abscissa values so that the transformed values are $(-2, y_{-2})$, $(-1, y_{-1})$ $(0, y_0)$, $(1, y_1)$, and $(2, y_2)$. What then is $\bar{y}(x_0)$, the smoothed least squares value of y_0? Let the approximating parabola be given by equation (1):

$$\bar{y} = a_0 + a_1 X_T + a_2 X_T^2 \tag{1}$$

where X_T is the transformed **X** value.

Thus, the problem has been reduced to minimizing S, with respect to the coefficients a_0, a_1, and a_2, where S is given by equation (2)

$$S = \sum_{i=2}^{2} \left(y_i - a_0 - a_1 X_{Ti} - a_2 X_{Ti}^2\right)^2 \tag{2}$$

Hence, the normal equations are

$$\frac{\partial S}{\partial a_0} = \sum_{i=-2}^{2} \left[-2\left(y_i - a_0 - a_1 X_{Ti} - a_2 X_{Ti}^2\right)\right] = 0$$

$$\frac{\partial S}{\partial a_1} = \sum_{i=-2}^{2} \left[-2\left(X_{Ti} y_i - a_0 X_{Ti} - a_1 X_{Ti}^2 - a_2 X_{Ti}^3\right)\right] = 0$$

$$\frac{\partial S}{\partial a_2} = \sum_{i=-2}^{2} \left[-2\left(X_{Ti}^2 y_i - a_0 X_{Ti}^2 - a_1 X_{Ti}^3 - a_2 X_{Ti}^4\right)\right] = 0$$

Since the following relations hold,

$$\sum_{i=-2}^{2} X_{Ti} = 0 \qquad \sum_{i=-2}^{2} X_{Ti}^2 = 10$$

$$\sum_{i=-2}^{2} X_{Ti}^3 = 0 \qquad \sum_{i=-2}^{2} X_{Ti}^4 = 34$$

we may simplify the normal equation in the following manner:

$$\frac{\partial S}{\partial a_0} = \sum_{i=-2}^{2} (y_i) - 5a_0 - 10a_2 = 0$$

$$\frac{\partial S}{\partial a_1} = \sum_{i=-2}^{2} (X_{Ti} y_i) - 10a_1 = 0$$

$$\frac{\partial S}{\partial a_2} = \sum_{i=-2}^{2} \left(X_{Ti}^2 y_i\right) - 10a_0 - 34a_2 = 0$$

The value of the parabolic equation (1) obtained at $X_T = 0$ is a_0; therefore, one may solve the system of preceding equations for a_0:

$$-3.4\left[(y_{-2} + y_{-1} + y_0 + y_1 + y_2) - 5a_0 - 10a_2\right] = 0$$

$$(4y_{-2} + y_{-1} + y_1 + 4y_2) - 10a_0 - 34a_2 = 0$$

Adding the two previous equations and solving for a_0 gives the following expression:

$$\bar{y}(\text{at } x_0) = a_0 = \frac{1}{35}(-3y_{-2} + 12y_{-1} + 17y_0 + 12y_1 - 3y_2)$$

A smoothing of this variety with endpoint corrections (ref. 6) is performed on the log counts prior to the normalization of the data and the spline fitting of the results.

Once the results of measurements for several gain settings have been normalized to the same gain setting and merged into the same data set, a least squares weighted exponential three point smoothing is performed on the merged data set (ref. 7). The mid-

point is weighted by unity and the two surrounding points are weighted by the ratio of their errors to the central point error. This final smoothing produces a smooth set of data suitable for spline fitting and interpolation.

SPLINE INTERPOLATION

Once the data points have been obtained in a suitable form, normalized to same gain setting, and merged, the data must be interpolated at arbitrary energy points to provide input data to a spectrum unfolding code, such as FERDOR. A spline interpolation has been chosen for this code, so as not to introduce extraneous undulations into the proton recoil spectrum, as a least squares polynomial fitting of the data might.

A spline is a device used in drafting for drawing smooth curves; it consists of a flexible strip of wood or metal which can be bent to pass through any points. If the spline is uniformly flexible and if it lies along the plane curve $y = F(x)$, its potential energy is

$$\lambda \int \left[F''(x)\right]^2 \left[1 + F'(x)^2\right]^{-3/2} dx \tag{3}$$

where λ is a proportionality constant. The form of the spline is such as to minimize equation (3) consistent with the constraints imposed by the points it passes through. A mathematical spline is defined in the same manner except that the $F'(x)^2$ term is removed.

A cubic spline has the following three features:

(1) If there exists a set of numbers $a = x_0 < x_1 \ldots < x_N = b$ and a set of corresponding y values $y_0, y_1, \ldots, y_N$, the cubic spline $F(x)$ satisfies $F(X_K) = y_K$ $(K = 0, 1, \ldots, N)$.

(2) $F(X)$ and $F'(X)$ are continuous (X_0, X_N).

(3) $\int_{X_0}^{X_N} \left[F''(X)\right]^2 dx$ exists in the interval (X_0, X_N) and is minimized subject to the two features previously given. The cubic spline is constructed in the following manner (refs. 8 and 9): Let $S_j = X_{j+1} - X_j$ and $t_j = F''(X_j)$ in the interval (X_j, X_{j+1}). $F(X)$ is then

$$F(X = \frac{\left[(X - X_j)y_{j+1} + (X_{j+1} - X)y_j\right]}{S_j}$$

$$- (X - X_j)(X_{j+1} - X) \frac{\left[(S_j + X_{j+1} - X)t_j + (S_j + X - X_j)t_{jH}\right]}{6S_j}$$

The condition that $F'(X)$ is continuous at X_j is expressed as

$$F'(X_j) = \frac{(y_{j+1} - y_j)}{S_j} - \frac{(2t_j + t_{j+1})S_j}{6} = \frac{(y_j - y_{j-1})}{S_{j-1}} + \frac{(2t_j + t_{j-1})S_{j-1}}{6}$$

$$(j = 2, 3, \ldots, N - 1) \qquad t_N = t_1 = 0.5$$

The values of $t_2, \ldots, t_{n-1}$ are determined from the solution of the matrix equation

$$\underline{\underline{A}}\vec{t} = \vec{u}$$

where

$$\underline{\underline{A}} = \begin{bmatrix} 2(S_1 + S_2) & S_2 & & & \\ S_2 & 2(S_2 + S_3) & S_3 & & \\ & S_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \\ & & & & 2(S_{N-2} + S_{N-1}) \end{bmatrix}$$

and the j^{th} element of $\vec{u}$ is given by

$$u_j = 6(y_{j+1} - y_j)/S_j - 6(y_j - y_{j-1})/S_{j-1}$$

The coding details are given in reference 9.

The spline interpolation thus provides a smooth interpolating function, without the difficulties introduced by polynomial curve fitting; for example, polynomial curve fitting often introduces extraneous detail into the data, undulations not found in the data (ref. 10).

EXTRAPOLATION OF DATA

The values at which interpolated data are required are fixed in the FERDOR input. At times the experimental values do not span the energy region required by FERDOR. Thus, the code PREJUD has provision for both low-energy and high-energy extrapolation. In both regions (low and high energy) weighted least squares exponential curve fits have been used (ref. 7).

For the low-energy extrapolation the first three low-energy proton recoil counts are weighted 10.0, 1.0, and 0.1 and an exponential is fitted in the weighted least squares manner and the FERDOR low-energy points are thus extrapolated.

The high-energy extrapolation uses the last five points with more than 25 counts per point and weights them 1, 2, 4, 8, and 16. One may easily change the value to a still smaller number of counts or larger as noted in the comment cards. Again a weighted least squares exponential fit is performed and used to extrapolate the data.

The extrapolations join the spline approximation in a smooth manner because of the rather large weights given end points.

ERROR ANALYSIS

As the FERDOR code requires not only counts, at previously decided upon energies, but count errors as well, the code PREJUD computes the error associated with each energy point prior to normalization. It computes the square root of the number of counts as the standard deviation of each point. If the data are normalized, the errors are normalized as well.

In order to ascertain the error of one of the interpolated points, the code determines the errors of the two surrounding points and uses a linear interpolation of the errors.

The errors associated with the low-energy extrapolation are in the same ratio as the lowest energy data point. That is, if the lowest energy data point is 10 000 counts with an error of 100 counts, an extrapolated point of 50 000 counts would have an error of 500 counts.

The errors associated with the high-energy extrapolation are the same as the error of the last point used in the extrapolation. That is, if the last point used is 36 counts with an error of 6 counts, an extrapolated point of 3 counts will be assigned an error of 6.

RESULTS

Example of Use of PREJUD

A Pu-Be source has been measured with a 1- by 1-inch (2.54- by 2.54-cm) liquid

organic scintillator operated at three gain settings, with correction for background. A separation of the proton recoil spectrum has been performed manually. Furthermore the input for FERDOR has been interpolated by hand. This proton-recoil spectrum has been unfolded by FERDOR to obtain a neutron spectrum. The same data have been treated by the code PREJUD and unfolded by FERDOR. The results of the manual-FERDOR and PREJUD-FERDOR are compared in figure 3. The unfolded neutron spectra from these two analyses are shown as two cross-hatched regions one standard deviation about their respective mean spectra.

Discussion of Example Problem

It is found that the two spectra overlap (are statistically equivalent) over the range of neutron energies from about 1 to 10 MeV. Given more analyzer channels in the low-energy region and adequate counting statistics in the high-energy region, the results of hand analyses and PREJUD-FERDOR unfoldings should match over the entire energy range.

The reason for the rather spread of the PREJUD-FERDOR spectrum is the rather conservative estimate of error; that is, PREJUD uses a linear interpolation of error values. The error values estimated in hand analyses are apparently associated with the smaller error of the two surrounding measured points.

The hand calculation represents on the order of at least 1 man day of work; whereas the execution time of PREJUD for this case was 0.54 minute on the Lewis IBM 7094-II.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, October 21, 1970,
129-02.

APPENDIX A

INPUT FORMAT FOR PREJUD

Card 1 Card columns 2 to 80 - Identification any valid Hollerith symbols

Card 2 Card columns 1 to 5 - format I5, number of different gain settings fixed point number from 1 to as many as 4

Card 3 Card columns 1 to 30 - format (2F10.5, I5, I5), gain setting information pertaining to following set of data:

Columns 1 to 10 (F10.5)	Columns 11 to 20 (F10.5)
M	B

where energy of midpoint of each energy bin associated with this gain setting is $E_{midpoint}$ = M*row number + B (in MeV β's),

Columns 21 to 25 (I5)	Columns 26 to 30 (I5)
Number of rows ignored	Number of columns ignored

Card 4+ Column 2 - namelist format, $FESS_IA=(Number of counts, row 1, column 1); then neglecting card column 1, all counts on 64 by 64 analyzer for this gain setting, by columns, counts separated by commas

Card 5+ Column 2 - namelist format, $FESS_IA=Same but background for this gain setting

Repeat cards 3, 4+, 5+ for other gain settings. If more than one run is to be performed, start new set with card 1 and same deck format.

APPENDIX B

PROGRAM LISTING

```
C         PREJUD PROGRAM   MAIN DECK
      COMMON IA
      DIMENSION E(64,4),IA(64,64),   NSUM1(64),NSUM2(64),MINR(64),MINB(6
     14),NSB1(64),NSB2(64),X1(64),Y1(64),ER1(64),X2(64),Y2(64),ER2(64),X
     23(64),Y3(64),ER3(64),X4(64),Y4(64),ER4(64),YS1(64),YS2(64),YS3(64)
     3,YS4(64),XINT(256),YINT(256),YMNT(256),ERRO(256),YTEMP(256),XTEMP(
     4256),Z(256),NR(64)
    1 READ(5,200)
      WRITE(6,205)
      WRITE(6,200)
      READ(5,202)NOFD
      IF(NOFD.GT.4) GO TO 1000
      GO TO 1005
 1000 WRITE(6,201)
      STOP
 1005 DO 3000 I = 1,NOFD
      READ(5,215)XM,B,ICHN,IZE
  215 FORMAT(2F10.5,I5,I5)
      DO 1010 J = 1,ICHN
 1010 E(J,I) = 0.0
      ISTAR1 = ICHN + 1
      DO 1015 J = ISTAR1,64
 1015 E(J,I) = B + XM*FLOAT(J)
      CALL RD
      MI = 0
      DO 1025 I1 = ISTAR1,64
      DO 1020 J1 = 1,64
 1020 NR(J1) = IA(I1,J1)
      CALL DSUM(NR,NSUM1,NSUM2,MI,I1,IZE)
 1025 MINR(I1) = MI
C  AT THIS POINT IN COMPUTATIONS THE MINIMUM COLUMN NOS. ARE STORED IN
C  MINR(64) FOR THIS GAIN SETTING.
      CALL RD
      MR = 0
      DO 1045 I1 = ISTAR1,64
      DO 1040 J1 = 1,64
 1040 NR(J1) = IA(I1,J1)
      CALL DSUM(NR,NSB1,NSB2,MR,I1,IZE)
 1045 MINB(I1) = MR
C    AT THIS POINT IN COMPUTATIONS THE MINIMUM COLUMN NOS. OF THE BACK-
C     GROUND ARE STORED IN   MINB(64) FOR THIS GAIN SETTING.
      GO TO (2000,2200,2400,2600),I
C  FIRST SET OF DATA
 2000 DO 2005 I2= 1,64
      X1(I2)= E(I2,I)
      Y1(I2) = NSUM2(I2)-NSB2(I2)
 2005 ER1(I2) = SQRT(ABS(Y1(I2)))
      DO 2010 I3 = ISTAR1,64
      IZ = I3 - ICHN
      X1(IZ) = X1(I3)
      Y1(IZ) = Y1(I3)
```

```
2010 ER1(IZ) = ER1(I3)
2011 ICLEAN = 65 - ICHN
     DO 2012 ICLE = ICLEAN,64
     X1(ICLE) = 0.0
     Y1(ICLE) = 0.0
2012 ER1(ICLE) = 0.0
   4 FORMAT(1H1,6X,42HFIRST SET OF DATA ENERGY,COUNTS,AND ERROR.)
     WRITE(6,4)
     WRITE(6,3)(X1(IK),Y1(IK),ER1(IK),IK=1,64)
   3 FORMAT(3G16.8)
2015 IF(Y1(1).GT.0.0) GO TO 2030
     DO 2020 I4 = 1,63
     X1(I4) = X1(I4+1)
     Y1(I4) = Y1(I4+1)
2020 ER1(I4) = ER1(I4 + 1)
     GO TO 2015
2030 NOF1 = 0
     DO 2035 I5 = 1,64
     IF(Y1(I5).GT.0.0) NOF1 = NOF1 + 1
     IF(Y1(I5).LE.0.0) GO TO 2040
2035 CONTINUE
2040 NP = NOF1 + 1
     DO 2045 I6 = NP,64
     X1(I6) = 0.0
     Y1(I6) = 0.0
2045 ER1(I6) = 0.0
     GO TO 2000
2200 DO 2205 I12 = 1,64
     X2(I12) =E(I12,I)
     Y2(I12) = NSUM2(I12) - NSB2(I12)
2205 ER2(I12) = SQRT(ABS(Y2(I12)))
     DO 2210 I13=ISTAR1,64
     IZ1 = I13 -ICHN
     X2(IZ1) = X2(I13)
     Y2(IZ1) = Y2(I13)
2210 ER2(IZ1) = ER2(I13)
2211 ICLEAN = 65 - ICHN
     DO 2212 ICLE = ICLEAN,64
     X2(ICLE) = 0.0
     Y2(ICLE) = 0.0
2212 ER2(ICLE) = 0.0
   5 FORMAT(1H1,6X,43HSECOND SET OF DATA ENERGY,COUNTS,AND ERROR.)
     WRITE(6,5)
     WRITE(6,3)(X2(IK),Y2(IK),ER2(IK),IK = 1,64)
2215 IF(Y2(1).GT.0.0) GO TO 2230
     DO 2220 I14 = 1,63
     X2(I14) = X2(I14 + 1)
     Y2(I14) = Y2(I14 + 1)
2220 ER2(I14) = ER2(I14 + 1)
     GO TO 2215
2230 NOF2 = 0
     DO 2235 I15 = 1,64
     IF(Y2(I15).GT.0.0) NOF2 = NOF2 + 1
     IF(Y2(I15).LE.0.0) GO TO 2240
2235 CONTINUE
2240 NP1 = NOF2 + 1
     DO 2245 I16 = NP1,64
     X2(I16) = 0.0
     Y2(I16) = 0.0
```

```
2245 ER2(I16) = 0.0
     GO TO 3000
2400 DO 2405 I22 = 1,64
     X3(I22) = E(I22,1)
     Y3(I22) = NSUM2(I22) - NSB2(I22)
2405 ER3(I22) = SQRT(ABS(Y3(I22)))
     DO 2410 I23 = ISTAR1,64
     IZ2 = I23 - ICHN
     X3(IZ2) = X3(I23)
     Y3(IZ2) = Y3(I23)
2410 ER3(IZ2) = ER3(I23)
2411 ICLEAN = 65 - ICHN
     DO 2412 ICLE = ICLEAN,64
     X3(ICLE) = 0.0
     Y3(ICLE) = 0.0
2412 ER3(ICLE) = 0.0
   6 FORMAT(1H1,6X,42HTHIRD SET OF DATA ENERGY,COUNTS,AND ERROR.)
     WRITE(6,6)
     WRITE(6,3)(X3(IK),Y3(IK),ER3(IK),IK=1,64)
2415 IF(Y3(1).GT.0.0) GO TO 2430
     DO 2420 I24 = 1,63
     X3(I24) = X3(I24 + 1)
     Y3(I24) = Y3(I24 + 1)
2420 ER3(I24) = ER3(I24 + 1)
     GO TO 2415
2430 NOF3 = 0
     DO 2435 I25 = 1,64
     IF(Y3(I25).GT.0.0) NOF3 = NOF3 + 1
     IF(Y3(I25).LE.0.0) GO TO 2440
2435 CONTINUE
2440 NP2 = NOF3 + 1
     DO 2445 I26 = NP2,64
     X3(I26) =0.0
     Y3(I26) =0.0
2445 ER3(I26) = 0.0
     GO TO 3000
2600 DO 2605 I32 = 1,64
     X4(I32) = E(I32,1)
     Y4(I32) = NSUM2(I32) - NSB2(I32)
2605 ER4(I32) = SQRT(ABS(Y4(I32)))
     DO 2610 I33 = ISTAR1,64
     IZ3 = I33 - ICHN
     X4(IZ3) = X4(I33)
     Y4(IZ3) = Y4(I33)
2610 ER4(IZ3) = ER4(I33)
2611 ICLEAN = 65 - ICHN
     DO 2612 ICLE = ICLEAN,64
     X4(ICLE) = 0.0
     Y4(ICLE) = 0.0
2612 ER4(ICLE) = 0.0
   7 FORMAT(1H1,6X,43HFOURTH SET OF DATA ENERGY,COUNTS,AND ERROR.)
     WRITE(6,7)
     WRITE(6,3)(X4(IK),Y4(IK),ER4(IK),IK=1,64)
2615 IF(Y4(1).GT.0.0) GO TO 2630
     DO 2620 I34 = 1,63
     X4(I34) = X4(I34 + 1)
     Y4(I34) = Y4(I34 + 1)
2620 ER4(I34) = ER4(I34 + 1)
     GO TO 2615
```

```
 2630 NOF4 = 0
      DO 2635 I35 = 1,64
      IF(Y4(I35).GT.0.0) NOF4 = NOF4 + 1
      IF(Y4(I35).LE.0.0) GO TO 2640
 2635 CONTINUE
 2640 NP3 = NOF4 + 1
      DO 2645 I36 = NP3,64
      X4(I36) = 0.0
      Y4(I36) = 0.0
 2645 ER4(I36) = 0.0
      GO TO 3000
 3000 CONTINUE
      NST = NOF1
C    HERE THE FIRST   SET OF DATA IS SMOOTHED
 3010 CALL SM(Y1,NOF1,YS1)
C     CALL SSM(YS1,NOF1)  CARDS MAY BE ADDED OR TAKEN OUT TO FURTHER
C    SMOOTH DATA,OR INCREASE FIDELITY TO ORIGINAL DATA.  HERE 2 HAVE
C    BEEN USED.
      CALL SSM(YS1,NOF1)
      CALL SSM(YS1,NOF1)
      IF(NST.EQ.NOF1) GO TO 3020
      ND = NOF1 + 1
      DO 3015 IM = ND,64
      X1(IM) = 0.0
      YS1(IM) = 0.0
 3015 ER1(IM) = 0.0
 3020 IF((NOFD - 1).LE.0) GO TO 3200
      NST = NOF2
C    HERE THE SECOND  SET OF DATA IS SMOOTHED
 3030 CALL SM(Y2,NOF2,YS2)
C     CALL SSM(YS2,NOF2)  CARDS MAY BE ADDED OR TAKEN OUT TO FURTHER
C    SMOOTH DATA,OR INCREASE FIDELITY TO ORIGINAL DATA.  HERE 2 HAVE
C    BEEN USED.
      CALL SSM(YS2,NOF2)
      CALL SSM(YS2,NOF2)
      IF(NST.EQ.NOF2) GO TO 3040
      ND = NOF2 + 1
      DO 3035 IM = ND,64
      X2(IM) = 0.0
      YS2(IM) = 0.0
 3035 ER2(IM) = 0.0
 3040 IF((NOFD - 2).LE.0) GO TO 3200
      NST = NOF3
C    HERE THE THIRD   SET OF DATA IS SMOOTHED
 3050 CALL SM(Y3,NOF3,YS3)
C     CALL SSM(YS3,NOF3)  CARDS MAY BE ADDED OR TAKEN OUT TO FURTHER
C    SMOOTH DATA,OR INCREASE FIDELITY TO ORIGINAL DATA.  HERE 2 HAVE
C    BEEN USED.
      CALL SSM(YS3,NOF3)
      CALL SSM(YS3,NOF3)
      IF(NST.EQ.NOF3) GO TO 3060
      ND = NOF3 + 1
      DO 3055 IM = ND,64
      X3(IM) = 0.0
      YS3(IM) = 0.0
 3055 ER3(IM) = 0.0
 3060 IF((NOFD - 3).LE.0) GO TO 3200
      NST = NOF4
```

```
C    HERE THE FOURTH  SET OF DATA IS SMOOTHED
 3070 CALL SM(Y4,NOF4,YS4)
C     CALL SSM(YS4,NOF4)  CARDS MAY BE ADDED OR TAKEN OUT TO FURTHER
C    SMOOTH DATA,OR INCREASE FIDELITY TO ORIGINAL DATA.  HERE 2 HAVE
C    BEEN USED.
       CALL SSM(YS4,NOF4)
       CALL SSM(YS4,NOF4)
       IF(NST.EQ.NOF4) GO TO 3200
       ND = NOF4 + 1
       DO 3075 IM = ND,64
       X4(IM) = 0.0
       YS4(IM) = 0.0
 3075 ER4(IM) = 0.0
       GO TO 3200
 3200 GO TO (3300,3400,3600,3800), NOFD
 3300 DO 3305 IPIC = 1,NOF1
       XINT(IPIC) = X1(IPIC)
       YINT(IPIC) = YS1(IPIC)
 3305 ERRO(IPIC) = ER1(IPIC)
       NALL = NOF1
       GO TO 4000
 3400 CSUB21 = CON(X1,YS1,NOF1,X2,YS2,NOF2)
    8 FORMAT(1H5,3X,53HNORMALIZATION CONSTANT OF SECOND DATA SET TO FIRS
     XT IS)
       WRITE(6,8)
       WRITE(6,6010) CSUB21
       DO 3405 IF21 = 1,NOF2
       YS2(IF21) = CSUB21*YS2(IF21)
 3405 ER2(IF21) = CSUB21*ER2(IF21)
    9 FORMAT(5X,26HNORMALIZED SECOND DATA SET)
       WRITE(6,9)
       WRITE(6,6010) (X2(I),YS2(I),ER2(I),I = 1,64)
       DO 3410 IRIC = 1,NOF1
       XINT(IRIC) = X1(IRIC)
       YINT(IRIC) = YS1(IRIC)
 3410 ERRO(IRIC) = ER1(IRIC)
       CALL MERGE(XINT,YINT,ERRO,NOF1,X2,YS2,ER2,NOF2,NALL)
   10 FORMAT(1H1,5X,33HMERGED FIRST AND SECOND DATA SETS)
       WRITE(6,10)
       WRITE(6,6010) (XINT(I),YINT(I),ERRO(I),I = 1,130)
       GO TO 4000
 3600 CSUB21 = CON(X1,YS1,NOF1,X2,YS2,NOF2)
       WRITE(6,8)
       WRITE(6,6010) CSUB21
       CSUB32 = CON(X2,YS2,NOF2,X3,YS3,NOF3)
   11 FORMAT(4X,53HNORMALIZATION CONSTANT OF THIRD DATA SET TO SECOND IS
     X)
       WRITE(6,11)
       WRITE(6,6010) CSUB32
       DO 3605 IF21 = 1,NOF2
       YS2(IF21) = CSUB21*YS2(IF21)
 3605 ER2(IF21) = CSUB21*ER2(IF21)
       WRITE(6,9)
       WRITE(6,6010)(X2(I),YS2(I),ER2(I),I=1,64)
       DO 3610 IF31 = 1,NOF3
       YS3(IF31) = CSUB32*CSUB21*YS3(IF31)
 3610 ER3(IF31) = CSUB32*CSUB21*ER3(IF31)
   12 FORMAT(5X,25HNORMALIZED THIRD DATA SET)
       WRITE(6,12)
```

```
      WRITE(6,6010)(X3(I),YS3(I),ER3(I),I=1,64)
      DO 3615 ISIC = 1,NOF1
      XINT(ISIC) = X1(ISIC)
      YINT(ISIC) = YS1(ISIC)
3615  ERRO(ISIC) = ER1(ISIC)
      CALL MERGE(XINT,YINT,ERRO,NOF1,X2,YS2,ER2,NOF2,NALL)
      WRITE(6,10)
      WRITE(6,6010) (XINT(I),YINT(I),ERRO(I),I=1,128)
      NOFM1 = NALL
      CALL MERGE(XINT,YINT,ERRO,NOFM1,X3,YS3,ER3,NOF3,NALL)
   13 FORMAT(1H5,6X,38HNORMALIZATION OF FIRST THREE DATA SETS)
      WRITE(6,13)
      WRITE(6,6010) (XINT(I),YINT(I),ERRO(I),I=1,196)
3620  GO TO 4000
3800  CSUB21 = CON(X1,YS1,NOF1,X2,YS2,NOF2)
      WRITE(6,8)
      WRITE(6,6010) CSUB21
      CSUB32 = CON(X2,YS2,NOF2,X3,YS3,NOF3)
      WRITE(6,11)
      WRITE(6,6010) CSUB32
      CSUB43 = CON(X3,YS3,NOF3,X4,YS4,NOF4)
   14 FORMAT(4X,53HNORMALIZATION CONSTANT OF FOURTH DATA SET TO THIRD IS
     X)
      WRITE(6,14)
      WRITE(6,6010) CSUB43
      DO 3805 IF21 = 1,NOF2
      YS2(IF21) = CSUB21 *YS2(IF21)
3805  ER2(IF21) = CSUB21 * ER2(IF21)
      WRITE(6,9)
      WRITE(6,6010)(X2(I),YS2(I),ER2(I),I = 1,64)
      DO 3810 IF31 = 1,NOF3
      YS3(IF31) = CSUB32*CSUB21*YS3(IF31)
3810  ER3(IF31) = CSUB32*CSUB21*ER3(IF31)
      WRITE(6,12)
      WRITE(6,6010)(X3(I),YS3(I),ER3(I),I = 1,64)
      DO 3815 IF41 = 1,NOF4
      YS4(IF41) = CSUB43*CSUB32*CSUB21*YS4(IF41)
3815  ER4(IF41) = CSUB43*CSUB32*CSUB21*ER4(IF41)
   15 FORMAT(5X,26HNORMALIZED FOURTH DATA SET)
      WRITE(6,15)
      WRITE(6,6010)(X4(I),YS4(I),ER4(I),I = 1,64)
      DO 3820 ITIC = 1,NOF1
      XINT(ITIC) = X1(ITIC)
      YINT(ITIC) = YS1(ITIC)
3820  ERRO(ITIC) = ER1(ITIC)
      CALL MERGE(XINT,YINT,ERRO,NOF1,X2,YS2,ER2,NOF2,NALL)
      WRITE(6,10)
      WRITE(6,6010)(XINT(I),YINT(I),ERRO(I),I = 1,128)
      NOFM1 = NALL
      CALL MERGE(XINT,YINT,ERRO,NOFM1,X3,YS3,ER3,NOF3,NALL)
      WRITE(6,13)
      WRITE(6,6010)(XINT(I),YINT(I),ERRO(I),I = 1,192)
      NOFM2 = NALL
      CALL MERGE(XINT,YINT,ERRO,NOFM2,X4,YS4,ER4,NOF4,NALL)
   16 FORMAT(1H5,6X,37HNORMALIZATION OF FIRST FOUR DATA SETS)
      WRITE(6,16)
      WRITE(6,6010)(XINT(I),YINT(I),ERRO(I),I = 1,256)
3825  GO TO 4000
```

```
C       EXPONENTIAL SMOOTHING OF DATA POINTS
 4000 NALLM = NALL - 1
      DO 4010 IZR = 2,NALLM
      DIMENSION EXSM(3),EYSM(3),EWSM(3)
      YMNT(1) = YINT(1)
      EXSM(1) = XINT(IZR - 1)
      EXSM(2) = XINT(IZR)
      EXSM(3) = XINT(IZR + 1)
      EYSM(1) = YINT(IZR - 1)
      EYSM(2) = YINT(IZR)
      EYSM(3) = YINT(IZR + 1)
      EWSM(1) = ERRO(IZR)/ERRO(IZR - 1)
      EWSM(2) = 1.0
      EWSM(3) = ERRO(IZR)/ERRO(IZR + 1)
 4010 YMNT(IZR) = WEXSMO(EXSM,EYSM,EWSM)
      YMNT(NALL) = YINT(NALL)
      NALL = NALL + 1
C       INTERPOLATION POINTS FOR FERDOR
      DATA(Z(I),I=1,256)/.0317,.0381,.0444,.0508,.0572,.0634,.0762,.0889
     X,.1015,.1205,.1395,.165,.1968,.2286,.2667,.3048,.3492,.3937,.4445,
     X.4953,.5524,.6096,.6668,.7239,.7810,.8382,.8954,.9525,.9525,1.016,
     X1.016,1.080,1.080,1.143,1.143,1.206,1.27,1.334,1.397,1.46,1.524,1.
     X5875,1.651,1.714,1.778,1.842,1.905,1.968,2.032,2.096,2.159.2.222,2
     X.286,2.350,2.413,2.476,2.54,2.667,2.794,2.921,3.048,3.175,3.302,3.
     X429,3.556,3.683,3.81,3.937,4.064,4.191,4.318,4.445,4.572,4.699,4.8
     X26,4.953,5.08,5.207,5.334,5.44,5.58,5.72,5.84,5.96,6.09,6.22,6.34,
     X6.54,6.73,6.92,7.11,7.30,7.49,7.68,7.88,8.07,8.26,8.44,8.64,8.82,9
     X.02,9.21,9.40,9.58,9.78,9.97,10.16,10.35,10.55,10.73,10.93,11.10,1
     X1.30,143*0.0/
C       SPLINE  FIT
 5000 NALL = NALL - 1
      CALL SPLN(XINT,YMNT,NALL,Z,YTEMP,113)
      CALL ERROR(XINT,ERRO,NALL,Z,XTEMP,113)
 5010 KAS = 1
 5015 IF(YTEMP(KAS).NE.0.0) GO TO 5030
C      LOW ENERGY EXTRAPOLATION
 5020 CALL EXTRA(XINT(1),YMNT(1),XINT(2),YMNT(2),XINT(3),YMNT(3),Z(KAS),
     1YTEMP(KAS),KAS)
      XTEMP(KAS) = YTEMP(KAS)*(ERRO(1)/YMNT(1))
      KAS = KAS + 1
      GO TO 5015
 5030 NIR = 1
      DO 5035 LINC = 1,256
      IF(YMNT(LINC).GT.25.0) NALL = LINC
      IF(YMNT(LINC).LE.25.0) GO TO 5036
 5035 CONTINUE
 5036 CONTINUE
      DO 5040 LR = 1,113
      IF(YTEMP(LR).GT.25.0) GO TO 5040
C    HIGH ENERGY EXTRAPOLATION
      CALL HIEXT(XINT(NALL-4),YMNT(NALL-4),XINT(NALL-3),YMNT(NALL-3),
     1XINT(NALL-2),YMNT(NALL-2),XINT(NALL-1),YMNT(NALL-1),XINT(NALL),YMN
     2T(NALL),Z(LR),YTEMP(LR),NIR)
      NIR = 2
      XTEMP(LR) = ERRO(NALL)
 5040 CONTINUE
 6000 WRITE(6,6010)(Z(I),YTEMP(I),XTEMP(I),I=1,113)
  200 FORMAT(80H
     1                              )
```

```
  201 FORMAT(19H TOO MANY DATA SETS)
  202 FORMAT(I5)
  205 FORMAT(1H1)
 6010 FORMAT(3G16.8)
 6020 PUNCH 200
      PUNCH 6030,(I,YTEMP(I),XTEMP(I),I=1,113)
 6030 FORMAT(25X,I3,22X,E10.3,10X,E10.3)
      GO TO 1
      END
```

```
$IBFTC LINER    DECK

      SUBROUTINE ERROR(X,Y,N,Z,XT,INT)
C     ERROR ESTIMATE
      DIMENSION X(256),Y(256),Z(256),XT(256)
      DO 40 I = 1,INT
      IF(Z(I).LT.X(1).OR.Z(I).GT.X(N)) XT(I) = 0.0
      IF(Z(I).LT.X(1).OR.Z(I).GT.X(N)) GO TO 40
      IF(Z(I).EQ.X(1)) XT(I) = Y(1)
      IF(Z(I).EQ.X(1)) GO TO 40
      IF(Z(I).EQ.X(N)) XT(I) = Y(N)
      IF(Z(I).EQ.X(N)) GO TO 40
      DO 30 J = 2,N
      IF(Z(I).LE.X(J).AND.Z(I).GE.X(J-1)) XT(I) = Y(J-1) +((Y(J)-Y(J-1))
     1/(X(J)-X(J-1)))*(Z(I)-X(J-1))
   30 CONTINUE
   40 CONTINUE
      RETURN
      END
```

```
$IBFTC WEXS     DECK

      FUNCTION WEXSMO(X,Y,W)
C    WEIGHTED EXPONENTIAL SMOOTHING OF DATA
      DIMENSION X(3),Y(3),W(3),H(3)
      DOUBLE PRECISION ZUCK,ZACK,ZORP,AKE,AKK,ARP,FARGO,APPLE
      NAG = 0
      DO 11 K = 1,3
      IF(Y(K))12,11,11
   12 Y(K) = ABS(Y(K))
      NAG = 1
   11 CONTINUE
      IF(NAG) 13,60,13
   13 WRITE(6,6)
    6 FORMAT(77HJNEGATIVE Y VALUES HAVE BEEN MADE POSITIVE IN CALCULATIO
     1NS BEYOND THIS LINE. ///)
   60 DO 130 I = 1,3
      H(I) = ALOG(Y(I))
  130 CONTINUE
      ZUCK = 0.0D0
      ZACK = 0.0D0
      ZORP = 0.0D0
      AKE  = 0.0D0
```

```
      AKK   = 0.0D0
      ARP   = 0.0D0
      DO 10 I = 1,3
      ZUCK = ZUCK + W(I)
      ZACK = ZACK + X(I)*W(I)
      ZORP = ZORP + H(I)*W(I)
      AKE = AKE + X(I)*W(I)
      AKK = AKK + W(I)*X(I)**2
   10 ARP = ARP + X(I)*H(I)*W(I)
      FARGO = ZUCK*AKK - ZACK*AKE
      IF(FARGO) 30,20,30
   20 WRITE(6,8)
    8 FORMAT(30H DIVISION BY ZERO NO SOLUTION.)
      RETURN
   30 APPLE = (ZORP*AKK - ZACK*ARP)/FARGO
      ALPHA = DEXP(APPLE)
      BETA = (ZUCK*ARP - ZORP*AKE)/FARGO
      WEXSMO = ALPHA*EXP(BETA*X(2))
      RETURN
      END
```

```
$IBFTC NOW      DECK

      SUBROUTINE SSM(X,N)
C     FIVE POINT SMOOTHING OF DATA POINTS
      DIMENSION X(64)
      DOUBLE PRECISION XD(64),XL(64)
      DO 10 I = 1,N
   10 XD(I) = X(I)
      IF(N.LT.5) GO TO 50
      DO 20 I = 1,N
      IF(XD(I).GT.0.0) XL(I) = DLOG10(XD(I))
   20 CONTINUE
      XD(1) = (XL(1)*69.0D0 + XL(2)*4.0D0 - XL(3)*6.0D0 + XL(4)*4.0D0 -
     1XL(5))/70.0D0
      XD(2) = (XL(1)*2.0D0 + XL(2)*27.0D0 + XL(3)*12.0D0 - XL(4)*8.0D0 +
     1 XL(5)*2.0D0)/35.0D0
      XD(N-1) = (XL(N-4)*2.0D0 - XL(N-3)*8.0D0 + XL(N-2)*12.0D0 + XL(N-1
     1)*27.0D0 + XL(N)*2.0D0)/35.0D0
      XD(N)=(-XL(N-4) + XL(N-3)*4.0D0 - XL(N-2)*6.0D0 + XL(N-1)*4.0D0 +
     1XL(N)*69.0D0)/70.0D0
      NM2 = N - 2
      DO 30 I = 3,NM2
   30 XD(I) = (-XL(I-2)*3.0D0 + XL(I-1)*12.0D0 + XL(I)*17.0D0 + XL(I+1)*
     112.0D0 - XL(I+2)*3.0D0)/35.0D0
      DO 40 I = 1,N
   40 X(I) = (10.0D0)**XD(I)
   50 RETURN
      END
```

```
$IBFTC NUN      DECK

      SUBROUTINE SM(X,N,XS)
C     FIVE POINT SMOOTHING OF DATA POINTS
      DIMENSION X(64),XS(64)
      DOUBLE PRECISION XD(64),XL(64)
      DO 10 I = 1,N
   10 XD(I) = X(I)
      IF(N.LT.5) GO TO 50
      DO 20 I = 1,N
      IF(XD(I).GT.0.0) XL(I) = DLOG10(XD(I))
   20 CONTINUE
      XD(1) = (XL(1)*69.0D0 + XL(2)*4.0D0 - XL(3)*6.0D0 + XL(4)*4.0D0 -
     1XL(5))/70.0D0
      XD(2) = (XL(1)*2.0D0 + XL(2)*27.0D0 + XL(3)*12.0D0 - XL(4)*8.0D0 +
     1 XL(5)*2.0D0)/35.0D0
      XD(N-1) = (XL(N-4)*2.0D0 - XL(N-3)*8.0D0 + XL(N-2)*12.0D0 + XL(N-1
     1)*27.0D0 + XL(N)*2.0D0)/35.0D0
      XD(N)=(-XL(N-4) + XL(N-3)*4.0D0 - XL(N-2)*6.0D0 + XL(N-1)*4.0D0 +
     1XL(N)*69.0D0)/70.0D0
      NM2 = N - 2
      DO 30 I = 3,NM2
   30 XD(I) = (-XL(I-2)*3.0D0 + XL(I-1)*12.0D0 + XL(I)*17.0D0 + XL(I+1)*
     112.0D0 - XL(I+2)*3.0D0)/35.0D0
      DO 40 I = 1,N
   40 XS(I) = (10.0D0)**XD(I)
      RETURN
   50 DO 60 I = 1,N
   60 XS(I) = X(I)
      RETURN
      END



$IBFTC MUB      DECK

      SUBROUTINE MERGE(XINT,YINT,ERRO,NOF1,X2,Y2,ER2,NOF2,IFINAL)
C      MERGE ROUTINE
      DIMENSION X1(256),Y1(256),ER1(256),X2(64),Y2(64),ER2(64),XINT(256)
     1,YINT(256),ERRO(256)
      IF(NOF1.GT.256)STOP
      DO 10 I = 1,NOF1
      X1(I) = XINT(I)
      Y1(I) = YINT(I)
      ER1(I) = ERRO(I)
      XINT(I) = 0.0
      YINT(I) = 0.0
   10 ERRO(I) = 0.0
   15 IFF = 1
      IMF = 1
      IFINAL = 1
   20 IF(X1(IFF).LT.X2(IMF)) GO TO 45
      IF(X1(IFF).EQ.X2(IMF)) GO TO 65
   25 XINT(IFINAL) = X2(IMF)
      YINT(IFINAL) = Y2(IMF)
      ERRO(IFINAL) = ER2(IMF)
   30 IF(IMF.EQ.NOF2) GO TO 35
      IMF = IMF + 1
```

```
      IFINAL = IFINAL + 1
      GO TO 20
   35 IF(IFF.EQ.NOF1) GO TO 40
      IFINAL = IFINAL + 1
      XINT(IFINAL) = X1(IFF)
      YINT(IFINAL) = Y1(IFF)
      ERRO(IFINAL) = ER1(IFF)
      IFF = IFF + 1
      GO TO 35
   40 IFINAL = IFINAL + 1
      XINT(IFINAL) = X1(NOF1)
      YINT(IFINAL) = Y1(NOF1)
      ERRO(IFINAL) = ER1(NOF1)
   44 RETURN
   45 XINT(IFINAL) = X1(IFF)
      YINT(IFINAL) = Y1(IFF)
      ERRO(IFINAL) = ER1(IFF)
   50 IF(IFF.EQ.NOF1) GO TO 55
      IFF = IFF + 1
      IFINAL = IFINAL + 1
      GO TO 20
   55 IF(IMF.EQ.NOF2) GO TO 60
      IFINAL = IFINAL + 1
      XINT(IFINAL) = X2(IMF)
      YINT(IFINAL) = Y2(IMF)
      ERRO(IFINAL) = ER2(IMF)
      IMF = IMF + 1
      GO TO 55
   60 IFINAL = IFINAL + 1
      XINT(IFINAL) = X2(NOF2)
      YINT(IFINAL) = Y2(NOF2)
      ERRO(IFINAL) = ER2(NOF2)
      RETURN
   65 IF(ER1(IFF).LT.ER2(IMF)) GO TO 70
      XINT(IFINAL) = X2(IMF)
      YINT(IFINAL) = Y2(IMF)
      ERRO(IFINAL) = ER2(IMF)
      IF(IFF.EQ.NOF1) GO TO 44
      IFF = IFF+ 1
      GO TO 30
   70 XINT(IFINAL) = X1(IFF)
      YINT(IFINAL) = Y1(IFF)
      ERRO(IFINAL) = ER1(IFF)
      IF(IMF.EQ.NOF2) IFF = IFF + 1
      IF(IMF.EQ.NOF2) GO TO 35
      IMF = IMF + 1
      GO TO 50
      END



$IBFTC PREXTR  DECK

      SUBROUTINE EXTRA(X1,Y1,X2,Y2,X3,Y3,XINT,YINT,KR2)
C     LOW ENERGY EXTRAPOLATION
      DIMENSION X(3),Y(3),W(3),H(3)
      DOUBLE PRECISION ZUCK,ZACK,ZORP,AKE,AKK,ARP,FARGO,APPLE
      IF(KR2.GT.1)GO TO 1000
```

```
      NAG = 0
      W(1) = 10.0
      W(2) = 1.0
      W(3) = 0.10
      X(1) = X1
      X(2) = X2
      X(3) = X3
      Y(1) = Y1
      Y(2) = Y2
      Y(3) = Y3
      DO 11 K = 1,3
      IF(Y(K))12,11,11
   12 Y(K) = ABS(Y(K))
      NAG = 1
   11 CONTINUE
      IF(NAG) 13,60,13
   13 WRITE(6,6)
    6 FORMAT(77HJNEGATIVE Y VALUES HAVE BEEN MADE POSITIVE IN CALCULATIO
     1NS BEYOND THIS LINE. ///)
   60 DO 130 I = 1,3
      H(I) = ALOG(Y(I))
  130 CONTINUE
      ZUCK = 0.0D0
      ZACK = 0.0D0
      ZORP = 0.0D0
      AKE  = 0.0D0
      AKK  = 0.0D0
      ARP  = 0.0D0
      DO 10 I = 1,3
      ZUCK = ZUCK + W(I)
      ZACK = ZACK + X(I)*W(I)
      ZORP = ZORP + H(I)*W(I)
      AKE = AKE + X(I)*W(I)
      AKK = AKK + W(I)*X(I)**2
   10 ARP = ARP + X(I)*H(I)*W(I)
      FARGO = ZUCK*AKK - ZACK*AKE
      IF(FARGO) 30,20,30
   20 WRITE(6,8)
    8 FORMAT(30H DIVISION BY ZERO NO SOLUTION.)
      RETURN
   30 APPLE = (ZORP*AKK - ZACK*ARP)/FARGO
      ALPHA = DEXP(APPLE)
      BETA = (ZUCK*ARP - ZORP*AKE)/FARGO
 1000 YINT = ALPHA * EXP(BETA*XINT)
      RETURN
      END



$IBFTC UP       DECK

      SUBROUTINE DSUM(NROW,NSUM1,NSUM2,MI,I,IZE)
C     MINIMUM FINDING ROUTINE
      DIMENSION NROW(64),NSUM1(64),NSUM2(64),A(64)
      MLST = MI
    2 FORMAT(1H ,9HDSUM1 ERR,I6)
      K = I
      WRITE(6,3)NROW,I
```

```
  3 FORMAT(16I7/16I7/16I7/16I7/I5)
    DO 10 II = 1,64
 10 A(II) = NROW(II)
    IF(MI.EQ.0) GO TO 12
    IF(MI.EQ.IZE) GO TO 60
    GO TO 100
 12 IF(A(IZE+1).GT.100.0.OR.A(IZE+2).GT.100.0) GO TO 20
    IF(A(IZE+3).GT.100.0.OR.A(IZE+4).GT.100.0) GO TO 20
    GO TO 60
 20 IA = 2 + IZE
 25 IF(A(IA-1).GT.A(IA).AND.A(IA).LT.A(IA+1)) GO TO 30
    IF(A(IA-2).GT.A(IA).AND.(A(IA).EQ.A(IA-1).AND.A(IA).LT.A(IA+1))) G
   1O TO 30
    IF(((A(IA-3).GT.A(IA).AND.A(IA).EQ.A(IA-1)).AND.A(IA).EQ.A(IA-2)).
   1AND.A(IA).LT.A(IA+1)) GO TO 30
    IF(A(IA).EQ.0.0) GO TO 40
 26 IA = IA + 1
    IF(IA.GE.64)WRITE(6,2)IA
    GO TO 25
 30 IF(A(IA).GE.(A(IA+1)-3.0*SQRT(A(IA+1))).OR.(A(IA).GT.A(IA+2))) GO
   XTO 35
    GO TO 40
 35 IF((A(IA+2)-A(IA)).LT.0.0)GO TO 26
    IF((A(IA+3)-A(IA)).LT.0.0)GO TO 26
    IF((A(IA+4)-A(IA)).LT.0.0)GO TO 26
 40 ISUM1 = 0
    ISUM2 = 0
    IB = IA -1
    IC = IA + 1
    DO 45 IR = 1,IB
 45 ISUM1 = ISUM1 + NROW(IR)
    ISUM1 = ISUM1 + NROW(IA)/2
    ISUM2 = NROW(IA)/2
    DO 50 IS = IC,64
 50 ISUM2 = ISUM2 + NROW(IS)
    IF(NROW(IA).GT.ISUM2/ 7) ISUM2 = 0
    IF(NROW(IA).GT.ISUM2/ 7) ISUM1 = 0
    IF(NROW(IA).GT.ISUM2/ 7) IA = 0
    IF( ISUM2       .EQ.0) GO TO 51
    IF((ISUM1/ISUM2).LE.10000) GO TO 51
    IA = 0
    ISUM1 = 0
    ISUM2 = 0
 51 MI = IA
    NSUM1(K) = ISUM1
    NSUM2(K) = ISUM2
7000 FORMAT(5X,24HMINIMUM COLUMN ABOVE IS ,I2)
    WRITE(6,7000)MI
    RETURN
 60 IF(AMAX1(A(IZE+1),A(IZE+2),A(IZE+3)).EQ.0.0) GO TO65
    GO TO 75
 65 IA = IZE
    ISUM1 = 0
    ISUM2 = 0
    DO 70 IZ = 1,64
 70 ISUM2 = ISUM2 + NROW(IZ)
    GO TO 51
 75 IA = IZE + 3
```

```
   76 IF(A(IA).EQ.0.0.AND.A(IA-1).EQ.0.0)GO TO 80
      IA = IA + 1
      IF(IA.EQ.64)WRITE(6,2)IA
      GO TO 76
   80 IA = IA-1
      ISUM1 = 0
      ISUM2 = 0
      DO 85 IZZ = 1,IA
   85 ISUM1 = ISUM1 + NROW(IZZ)
      DO 90 IZZ1=IA,64
   90 ISUM2 = ISUM2 + NROW(IZZ1)
      GO TO 51
  100 IF(MI-5.GT.IZE)IA =MI - 5
      IF(MI-5.GT.IZE)GO TO 25
      GO TO 20
      END
```

```
$IBFTC ZING     DECK

      FUNCTION CON(X,Y,N,XF,YF,M)
C     NORMALIZATION CONSTANT ROUTINE
      DIMENSION X(256),Y(256),X1(256),Y1(256),YS(256),XF(64),YF(64)
      DO 10 I = 1,64
      X1(I) = XF(I)
      Y1(I) = YF(I)
   10 YS(I) = 0.0
      CALL SPLN(X,Y,N,X1,YS,M)
      CONF = 0.0
      DIV = 0.0
      DO 20 I = 1,64
      IF(YS(I).NE.0.0)DIV = DIV+ 1.0
      IF(YS(I).NE.0.0) CONF = CONF + YS(I)/Y1(I)
   20 CONTINUE
      CON = CONF/DIV
      RETURN
      END
```

```
$IBFTC DUB      DECK

      SUBROUTINE RD
C     READ ROUTINE
      COMMON IA
      DIMENSION IA(64,64)
      NAMELIST/FESS/IA
    1 READ(5,FESS)
      RETURN
      END
```

```
$IBFTC POSTEX  DECK

      SUBROUTINE HIEXT(X1,Y1,X2,Y2,X3,Y3,X4,Y4,X5,Y5,XINT,YINT,K88)
C     HIGH ENERGY EXTRAPOLATION  ROUTINE
      DIMENSION X(5),Y(5),W(5),H(5)
      DOUBLE PRECISION ZUCK,ZACK,ZORP,AKE,AKK,ARP,FARGO,APPLE
      IF(K88.GT.1) GO TO 1000
      NAG = C
      W(1) = 1.0
      W(2) = 2.0
      W(3) = 4.0
      W(4) = 8.0
      W(5) = 16.00
      X(1) = X1
      X(2) = X2
      X(3) = X3
      X(4) = X4
      X(5) = X5
      Y(1) = Y1
      Y(2) = Y2
      Y(3) = Y3
      Y(4) = Y4
      Y(5) = Y5
      DO 11 K = 1,5
      IF(Y(K))12,11,11
   12 Y(K) = ABS(Y(K))
      NAG = 1
   11 CONTINUE
      IF(NAG) 13,60,13
   13 WRITE(6,6)
    6 FORMAT(77HJNEGATIVE Y VALUES HAVE BEEN MADE POSITIVE IN CALCULATIO
     1NS BEYOND THIS LINE. ///)
   60 DO 130 I = 1,5
      H(I) = ALOG(Y(I))
  130 CONTINUE
      ZUCK = 0.0D0
      ZACK = 0.0D0
      ZORP = 0.0D0
      AKE  = 0.0D0
      AKK  = 0.0D0
      ARP  = 0.0D0
      DO 10 I = 1,5
      ZUCK = ZUCK + W(I)
      ZACK = ZACK + X(I)*W(I)
      ZORP = ZORP + H(I)*W(I)
      AKE = AKE + X(I)*W(I)
      AKK = AKK + W(I)*X(I)**2
   10 ARP = ARP + X(I)*H(I)*W(I)
      FARGO = ZUCK*AKK - ZACK*AKE
      IF(FARGO) 30,20,30
   20 WRITE(6,8)
    8 FORMAT(30H DIVISION BY ZERO NO SOLUTION.)
      RETURN
   30 APPLE = (ZORP*AKK - ZACK*ARP)/FARGO
      ALPHA = DEXP(APPLE)
      BETA = (ZUCK*ARP - ZORP*AKE)/FARGO
 1000 YINT = ALPHA * EXP(BETA*XINT)
      RETURN
      END
```

```
$IBFTC SCRE     DECK

      SUBROUTINE SPLN(X,Y,N,Z,YINT,MAX)
C     SPLINE ROUTINE
      DIMENSION X(256),Y(256),Z(256),YINT(256)
      DOUBLE PRECISION S(256),A(256),B(256),C(256),F(256),W(256),SB(256)
     1,G(256),EM(256),DYDX(256),D2YDX(256),CURV(255),RAD(256),SJM(256)
      DO 10 I = 2,N
   10 S(I) = X(I) - X(I-1)
      NO = N - 1
      IF(2.GT.NO) GO TO 25
      DO 20 I = 2, NO
      A(I) = S(I)/6.0
      B(I) = (S(I)+S(I+1))/3.0
      C(I) = S(I+1)/6.0
   20 F(I) = (Y(I+1)-Y(I))/S(I+1)-(Y(I)-Y(I-1))/S(I)
   25 A(N) = -0.50
      B(1)= 1.0
      B(N)= 1.0
      C(1) = -0.50
      F(1) = 0.0
      F(N) = 0.0
      W(1) = B(1)
      SB(1) = C(1)/W(1)
      G(1) = 0.0
      DO 30 I = 2,N
      W(I) = B(I)-A(I)*SB(I-1)
      SB(I) = C(I)/W(I)
   30 G(I) = (F(I)-A(I)*G(I-1))/W(I)
      EM(N) = G(N)
      DO 40 I = 2,N
      K = N + 1 - I
   40 EM(K) = G(K) -SB(K)*EM(K+1)
      SUM(1) = 0.0
      DO 45 I = 2,N
   45 SUM(I)=SUM(I-1)+S(I)*(Y(I)+Y(I-1))/2.-S(I)**3*(EM(I)+EM(I-1))/24.
      DO 90 I = 1,MAX
      K = 2
      IF(Z(I)-X(1)) 60,50,70
   50 YINT(I)=Y(1)
      GO TO 86
   60 IF(Z(I).LT.X(1)) YINT(I) = 0.0
      IF(Z(I).LT.X(1)) GO TO 90
      GO TO 85
   65 IF(Z(I).GT.X(N)) YINT(I) = 0.0
      IF(Z(I).GT.X(N)) GO TO 90
      K = N
      GO TO 85
   70 IF(Z(I)-X(K)) 85,75,80
   75 YINT(I) = Y(K)
      GO TO 86
   80 K = K + 1
      IF(K-N) 70,70,65
   85 YINT(I)=EM(K-1)*(X(K)-Z(I))**3/6./S(K)+EM(K)*(Z(I)-X(K-1))**3/6.
     1/S(K)+(Y(K)/S(K)-EM(K)*S(K)/6.)*(Z(I)-X(K-1))+(Y(K-1)/S(K)-EM(K-1)
     2*S(K)/6.)*(X(K)-Z(I))
   86 DYDX(I)=-EM(K-1)*(X(K)-Z(I))**2/2.0/S(K)+EM(K)*(X(K-1)-Z(I))**2/2.
     10/S(K)+(Y(K)-Y(K-1))/S(K)-(EM(K)-EM(K-1))*S(K)/6.0
      D2YDX(I) = EM(K-1)*(X(K)-Z(I))/S(K)+EM(K)*(Z(I)-X(K-1))/S(K)
```

```
      CURV(I) = D2YDX(I)/(1.+DYDX(I)**2)**1.5
      RAD(I) = 1./CURV(I)
   90 CONTINUE
      RETURN
      END
```

REFERENCES

1. Birks, J. B.: The Theory and Practice of Scintillation Counting. Macmillan Co., 1964, pp. 219-227, pp. 392-394.

2. Brooks, F. D.: A Scintillation Counter with Neutron and Gamma-Ray Discriminators. Nucl. Inst. Methods, vol. 4, 1959, pp. 151-163.

3. Blue, J. W.; and Shook, D. F.: Another Scheme of Neutron and Gamma Identification Using Organic Scintillators. Bull. Am. Phys. Soc., vol. 13, no. 4, Apr. 1968, p. 559.

4. Burrus, W. R.; and Verbinski, V. V.: Recent Developments in the Proton-Recoil Scintillation Neutron Spectrometer. Proceedings of the Special Session on Fast Neutron Spectroscopy at the 1964 Winter Meeting of the American Nuclear Society. J. Greenburg, ed. Rep. ANS-SD-2, Am. Nucl. Soc., Feb. 15, 1965, pp. 148-188.

5. Clemons, Lester, Jr.; Wrights, Gilbert N.; and Shook, Donald F.: Monte Carlo Calculations of a Hemispherical-Duct Neutron-Streaming Experiment. NASA TM X-1971, 1970.

6. Nielsen, Kaj L.: Methods in Numerical Analysis. Macmillan Co., 1956, p. 289.

7. Semler, Thor T.: Program for Calculation of Neutron Ages from Experimental Activation Data. NASA TM X-1260, 1966, pp. 7, 23.

8. Walsh, J. L.; Ahlberg, J. H.; and Nilson, E. N.: Best Approximation Properties of the Spline Fit. J. Math. Mech., vol. 11, no. 2, 1962, pp. 225-234.

9. Katsanis, Theodore: Fortran Program for Spline Fit Curve. NASA TM X-1707, 1968.

10. Hartree, Douglas R.: Numerical Analysis. Clarendon Press, Oxford, 1952, pp. 92-94.

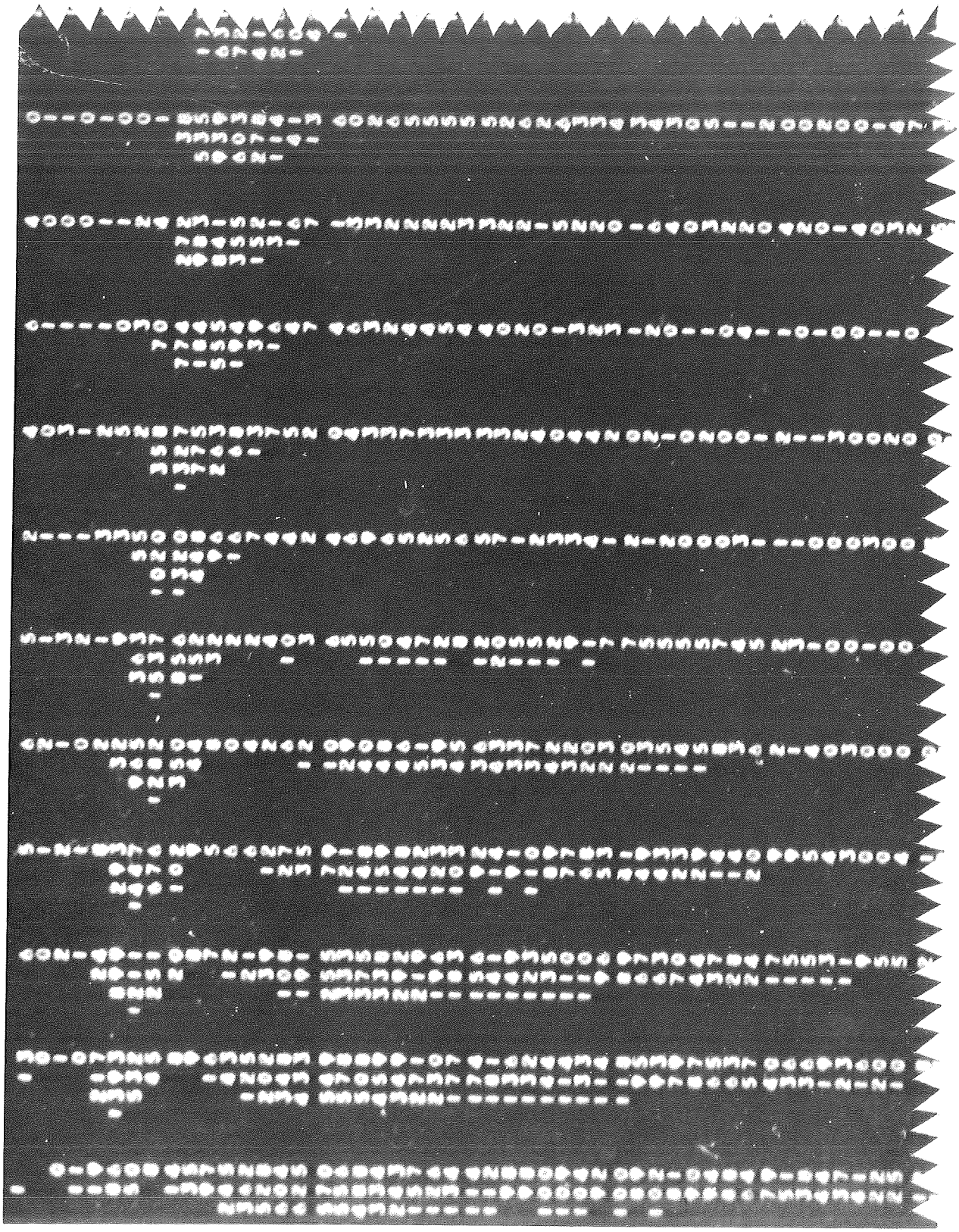


Figure 1. - Portion of output display of two-dimensional analyzer. (CS 46341)

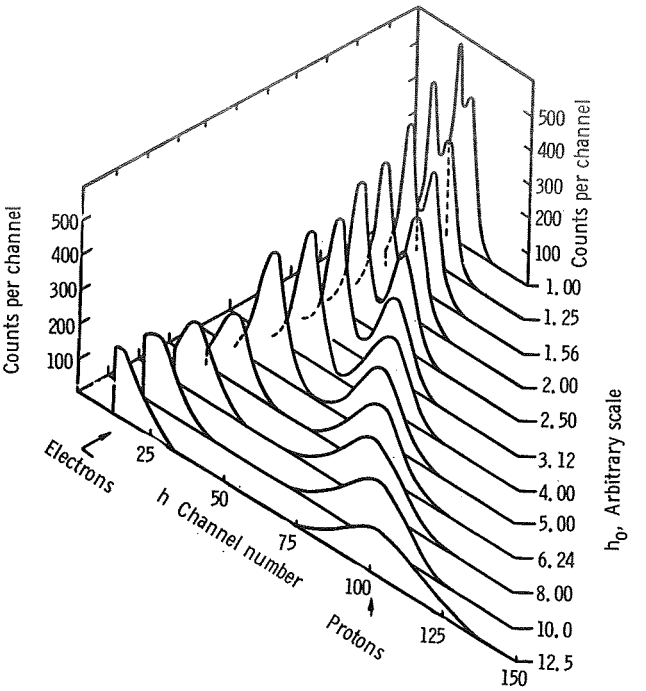


Figure 2. - Series of pulse height distributions obtained using stilbene and polonium-210 - beryllium source (ref. 2).

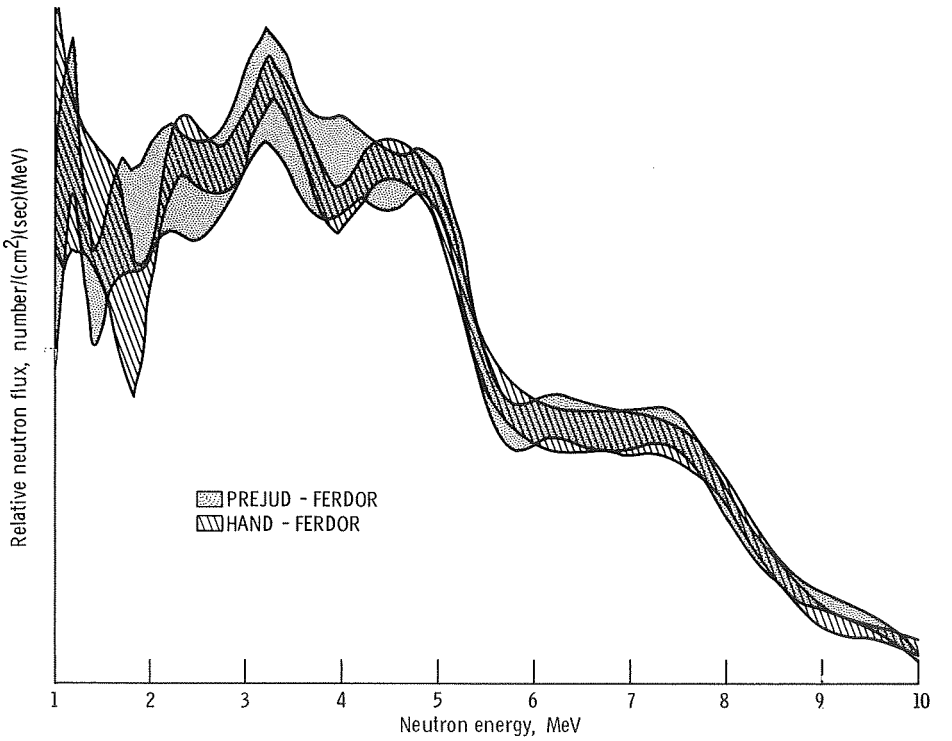


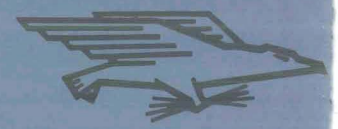
Figure 3. - Plutonium beryllium spectra unfolded by FERDOR input prepared manually and by PREJUD.

NASA-Langley, 1971 —— 22 E-5824

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL



POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546